

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №5

Выполнил:		Проверил:
студент группы ИУ5-51		преподаватель каф. ИУ5
Абросимова Надежда		Гапанюк Ю.Е.
Подпись и дата:		Подпись и дата:

Задание лабораторной работы.

В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также вам нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

Для сдачи должны быть:

1. Скрипт с подключением к БД и несколькими запросами.
2. Набор классов предметной области с привязкой к СУБД (класс должен уметь хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса).
3. Модели предметной области.
4. View для отображения списка сущностей.

Исходный код/

1. Обращение к БД из Python:

```
import pymysql
pymysql.install_as_MySQLdb()

db=pymysql.connect(
    host='localhost',
    user='dbuser',
    passwd='123',
    db='first_db'
)

c=db.cursor()

c.execute('SELECT *FROM books;')
entries=c.fetchall()

for e in entries:
    print(e)

c.close()
db.close()
```

2. Написание классов предметной области с соединением с БД.

```
import pymysql
pymysql.install_as_MySQLdb()

class Connection:
    def __init__(self, user, password, db, host='localhost'):
        # Сохраняем параметры соединения
        self.user = user
        self.password = password
        self.db = db
        self.host = host
        self._connection = None

    @property
    def connection(self):
        return self._connection

    def __enter__(self):
        self.connect()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()

    def connect(self):
        """
        Открытие соединения
        """
        if not self._connection:
            self._connection = pymysql.connect(
                host=self.host,
                user=self.user,
```

```

        passwd=self.password,
        db=self.db
    )

    def disconnect(self):
        """
        Закрытие соединения
        """
        if self._connection:
            self._connection.close()

class books:

    def __init__(self, db_connection, id, name, autor):
        self.db_connection = db_connection.connection
        self.id=id
        self.name = name
        self.autor = autor

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO Books (id, name, autor) VALUES (%s, %s, %s);", (self.id, self.name, self.autor))
        self.db_connection.commit()
        c.close()

con = Connection('dbuser', '123', 'first_db')

with con:
    book = books(con, '6', 'Книга Книжная', 'Автор Авторный')
    book.save()

```

3. Django ORM.

models.py:

```

from django.db import models

class BookModel(models.Model):
    class Meta:
        db_table = 'books'

    name = models.CharField(max_length=25)
    author = models.CharField(max_length=50)

    def __str__(self):
        return "{}, {}".format(self.name, self.author)

```

views.py:

```

from django.shortcuts import render

# Create your views here.
from django.views import View

from library.models import BookModel

class BookView(View):
    def get(self, request):
        books = BookModel.objects.all()
        data = {
            'books': books
        }
        return render(request, 'bookinfo.html', data)

```

bookinfo.html:

```

{% extends 'base.html' %}
{% block title %}Some data{% endblock %}
{% block body %}
    <ul>
        {% for book in books %}
            <li>
                Название - {{ book.name }}<br>
                Автор - {{ book.author }}.</li>
            {% endfor %}
        </ul>
    {% endblock %}

```